



Patent
Attorney's Docket No. P2380-505

11-15-02

leg for
reconsid

NP

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)
David G. OPSTAD et al.) Group Art Unit: 2672
Application No.: 09/306,888) Examiner: T. Havan
Filed: May 7, 1999) Confirmation No.: 4127
For: AUTOMATIC SYNTHESIS OF)
FONT TABLES FOR CHARACTER)
LAYOUT)

Rec'd
11/15/03

RECEIVED

REQUEST FOR RECONSIDERATION

NOV 14 2002

Assistant Commissioner for Patents
Washington, D.C. 20231

Technology Center 2600

Sir:

In response to the Official Action dated August 13, 2002, Applicants respectfully request reconsideration and withdrawal of the rejection of the claims.

At the outset, Applicants note with appreciation the indication on page 2 of the Office Action that claims 10, 14, 15, 21 and 28 contain allowable subject matter. However, Applicants submit that all claims 1-31 are allowable in view of the following remarks.

The Office Action includes a rejection of claims 1-9, 11-13, 16-20, 22-27 and 29-31 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,500,931 to Sonnenschein (hereinafter, *Sonnenschein*) in view of U.S. Patent No. 6,426,751 to Patel et al. (hereinafter, *Patel*). This rejection is respectfully traversed.

The *Sonnenschein* document is directed to an entirely different objective than the present application. In particular, *Sonnenschein* is concerned with solving problems involving operating systems which overload character codes in their code page architectures and fonts. According to *Sonnenschein*, one problem existing in many of these operating systems results from using a same character code to associate possibly different characters in more than one font. As a result, these systems cannot adequately detect, when a character is entered, whether that character exists in a font that is currently in use. This often leads to an operating

system associating a desired character (e.g., a character of a particular font intended by an author or editor of text) with an entirely different character (or none at all). Another problem caused by overloading of character codes discussed in *Sonnenschein* involves “garbling” of text when a client tries to apply a font style to a string of characters of different fonts.

To accomplish the above-mentioned objective, *Sonnenschein* discloses logic that “automatically styles a character with a new font style when the current font cannot display the character.” (See *Sonnenschein*, column 5, lines 1-3 (emphasis added).) More specifically, *Sonnenschein* describes a process in which characters of a text string are iteratively processed to determine whether a currently iterated character code can be mapped to a glyph. First, the character code in the current iteration is checked to see whether it can be mapped to a valid glyph in the currently set font. If so, the current font is associated with that character. Second, if the character code cannot be mapped to a valid glyph in the current font, the process then attempts to associate the current character code with one in the font associated with the current character (because “[e]very character in a text stream has a font style associated with it,” see *Sonnenschein*, column 5, lines 27-28). Next, if the first and second processes fail, all available fonts in the system are checked to find a font having a valid glyph. Finally, if all the above processes fail, the current character code is associated with a “predefined missing glyph system font” described in *Sonnenschein* as “a special font that is used for character substitution purposes only when no other font can display the character.” (See *Sonnenschein*, column 5, lines 31-34 and column 6, lines 3-5.)

In contrast to the system of *Sonnenschein*, the present invention is concerned with the fact that some of the tables that may be needed to properly display a font in accordance with certain line layout technologies may not be present, particularly in the case of older fonts. For example, as described on pages 9 and 10 of the application, one of the procedures that is carried out by a line layout processor comprises metamorphosis, in which an initial set of glyphs may be transformed into a different set of glyphs. In carrying out this procedure, the line layout processor refers to one or more tables in the font that identify the glyph changes that should occur. It may be the case, however, that some fonts do not contain these tables. The present invention addresses this situation, by providing a mechanism for automatically synthesizing the tables from the information contained in the font itself. For example, referring

to an exemplary embodiment depicted in Figure 7, the synthesizer employs data contained in some of the tables of a font, such as a glyph mapping table 47, to construct a font map 81. Then, using information in a mapping table 84, the synthesizer constructs a metamorphosis table 86, which can then be used by the line layout processor to properly display the font.

It is respectfully submitted that the *Sonnenschein* patent does not address this problem, and more importantly does not disclose the features of the present invention for automatically synthesizing one or more font tables.

The Office Action, at page 3, line 11 acknowledges that *Sonnenschein* does not teach creating a table. However, the Office Action points to column 4, lines 41-52 of *Sonnenschein*, and alleges that *Sonnenschein* teaches "a font table synthesizer which is responsive to the absence of a predetermined data table for creating and storing table [sic] on the basis of data contained in the font file." The undersigned has reviewed this portion of the *Sonnenschein* patent and cannot find any support whatsoever for this allegation. In fact, the cited portion of *Sonnenschein* discloses the following:

The first process is used when a character, or string of characters, is entered into a text string. If the character is missing from the font specified in the current typing style the process will choose a font that can display the character. For example, if the code for a "Σ" is inserted before "n/2" the process would automatically style the code with a font that could display the "Σ."

The second process is used when a client applies a font style change to a range of characters. The process intelligently applies the font to the selection. For example, applying the preferred embodiment of the invention, Chicago font to the characters "Σ n/2" produce "Σ n/2" (emphasis added).

Applicants assert that this portion of *Sonnenschein* does not disclose that if one or more required tables is missing, it can be automatically synthesized from data in the other tables. Rather, the *Sonnenschein* patent assumes that all of the necessary tables for generating a font will be present. Thus, with reference to claim 1, the patent does not disclose the step of "determining *whether* the font contains a predetermined data table ..." Similarly, it does not disclose the following step of "automatically synthesizing said data table ... *if* the font is determined not to contain said data." As set forth in this claim, the present invention operates to *detect* whether the tables necessary to lay out glyphs are present in a font, and then automatically *create*, or synthesize, a table if it is missing. The *Sonnenschein* patent does not

) Aug. 2007
in claim

disclose these operations, nor anything analogous thereto. In particular, there is no disclosure of the capability to determine that a required table is missing, and thereafter synthesize the table in response to such a determination.

These distinguishing characteristics of the present invention are brought out in the other pending claims, as well. For example, claim 11 recites "a font table synthesizer *which is responsive to the absence of a predetermined data table* for creating and storing said table on the basis of data contained in the font file." Claims 19 and 26 recite the steps of "*determining* whether the data table is present in a file containing the font; and *synthesizing* said table from data contained in said file *if the table is not present in the font file.*"

As another point of distinction, the particular manner in which a data table is constructed in a preferred embodiment of the present invention is significantly different from the process for generating a new font in the technique of *Sonnenschein*. For example, claim 6 recites the steps of "building a font map that contains information about individual glyphs in the font"; "determining relationships between items of information in the font map"; and "constructing a table which identifies said relationships." Similar recitations are found in claims 16, 22 and 29. With respect to this subject matter, the Office Action alleges that *Sonnenschein* discloses the steps of building a font map at column 5, lines 12-27. It is not apparent what information disclosed in this portion of *Sonnenschein* is considered to constitute a font map.

It is respectfully submitted that the *Sonnenschein* document neither discloses, nor otherwise suggests, any of these claimed features. It discloses techniques for styling a character with a new font style when a current font cannot display the character, but it does not disclose any mechanism for automatically synthesizing data determined to be missing from the font. In the processes described in *Sonnenschein*, if a character code cannot be mapped to a font, the processor simply looks among other fonts in the system for a match and replaces the font associated with the current character with the newly found one.

The Office Action acknowledges that *Sonnenschein* fails to disclose constructing a table. To make up for this shortcoming, the Office Action proposes modifying *Sonnenschein* with a teaching in *Patel* of creating a table for font layout. Applicants respectfully submit that the Office has failed to establish a *prima facie* case of obviousness because the proposed

combination of *Sonnenschein* and *Patel* does not teach the combination of every recited feature in each of the claims.

Patel is directed to a computer program having a front-end editable text file called “a feature file” that a user can edit and have a computer process to define changes to an existing font file or to create a font file. *Patel* describes the feature file as being an editable file that includes simple logical statements that can be set by a font editor in the form of high-level feature definition language, for example, in logic statements expressed in “English-like grammar.” (See *Patel*, column 3, lines 22-28.)

The Office Action alleges that *Patel* teaches constructing a table that “contains information on glyph positioning, glyph substitution, justification, and baseline positioning,” and that by “taking the combined teaching of *Sonnenschein* and *Patel* as a whole, it would have been obvious to combine the teaching of *Patel* to the system of *Sonnenschein* because doing so would have enabled processing fonts to improve font layout in a table format as noted in *Patel* (col. 1, lines 17-41; figs. 2 and 4-element 420) [sic]” (emphasis in original).

First, Applicants submit that the textual part of *Patel* cited in the Office Action is not directed to *constructing* a table. Rather, the cited portion of *Patel* the Office relies upon is describing conventional OpenType font layout tables. That is, *Patel*'s description is of tables that already exist in the OpenType font file, which is similar to the generalized description of font tables described in the background section of Applicants' specification. Applicants do note, however, that *Patel* does teach formation of tables (e.g., see *Patel*, column 4, lines 34-45), but this is in the context of processing the feature file to alter a font or to create a new font, and not because of any condition of determining that a font does not contain a predetermined data table, as recited in claim 1. Similarly, *Patel* does not teach a font table synthesizer which is *responsive to the absence of a predetermined data table* for creating and storing said table,” as recited in claim 11 or *determining whether the data table is present in a file containing a font, and synthesizing said table* from data contained in said file *if the table is not present in the file*,” as recited in claims 19 and 26.

The Office Action also identifies *Patel*'s Figures 2 and 4 as allegedly providing support in *Patel* for the proposed combination. Figure 2 shows a method that translates the internal representations derived from the feature file and creates subtables to be stored in the font file.

Figure 4 merely shows a computer system that implements the processing of a feature file to make changes in a font file. From these and other statements in the Office Action, it is not understood what nexus exists between the proposed combination of *Sonnenschein* and *Patel* and the claims. *Patel* teaches that fonts can be altered or created using a computer program having a front-end editable feature file. *Sonnenschein* is concerned with associating a character code with a font. As pointed out above, neither *Sonnenschein* nor *Patel* all the combinations of features claimed.

Moreover, even if one were to consider *arguendo* that one of ordinary skill in the art would have been led to make the proposed combination of *Sonnenschein* and *Patel*, this hypothetical combination would not teach the combinations of features recited in claims. At best, the proposed combination would appear to result in a system in which a user/editor can define a new font or alter a font by entering into a feature file declarative logic statements and then processing these statements, as taught in *Patel*, and one that would substitute, for a character, one font style for another font style if a current font could not display the character, as taught in *Sonnenschein*. For instance, when comparing claim 1 with this combination, it is clear that *Sonnenschein* and *Patel* do not teach a system that includes the step of "determining whether the font contains a predetermined data table that pertains to the layout of glyphs . . . automatically synthesizing said data table, based upon data contained in the font, *if the font is determined not to contain said data table*." Because the proposed combination does not teach this feature, claim 1 is not obvious because a *prima facie case* does not exist. As such, claim 1 is patentable. Other distinguishing features defined in independent claims 11, 16, 19, 26, and 29 were noted above. Applicants submit these claims also are patentable. To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). (See MPEP § 2143.03.)

Applicants also respectfully submit that in making the combination of *Patel* and *Sonnenschein*, the Office has picked elements from Applicants' claims, examined each in isolation, and then combined the elements without giving proper consideration to the claims as

a whole. It is a basic tenet of patent law that when applying 35 U.S.C. § 103 is that the claimed invention must be considered as a whole. MPEP § 2141.02 states that: “[i]n determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983); *Schenck v. Nortron Corp.*, 713 F.2d 782, 218 USPQ 698 (Fed. Cir. 1983).

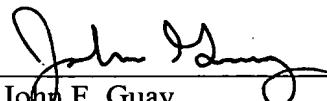
Because the differences between the proposed combination of *Sonnenschein* and *Patel* and the independent claims are clear, Applicants will not belabor a discussion of each of the remaining dependent claims. However, Applicants submit that the remaining dependent claims are allowable at least because they ultimately respectively depend from one the independent claims discussed above. Applicants also submit that these dependent claims set forth combinations of additional distinctions that are not taught in *Sonnenschein* and *Patel*.

Application No. 09/306,888
Attorney's Docket No. P2380-505
Page 8

Reconsideration and withdrawal of the rejection, and allowance of all pending claims
are respectfully requested.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: 
John F. Guay
Registration No. 47,248

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6696

Date: November 13, 2002



Patent
Attorney's Docket No. 001580-505

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of) **BOX NON-FEE AMENDMENT**
David G. OPSTAD et al.)
Application No.: 09/306,888) Group Art Unit: 2672
Filed: May 7, 1999) Examiner: Thu Thao Havan
For: AUTOMATIC SYNTHESIS OF FONT) Confirmation No.: 4127
TABLES FOR CHARACTER LAYOUT)
)
)
)
)
)
)
)
)
)
)
)
)
)
RECEIVED
NOV 14 2002
Technology Center 2600

AMENDMENT/REPLY TRANSMITTAL LETTER

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Enclosed is a Request for Reconsideration for the above-identified patent application.

[] A Petition for Extension of Time is also enclosed.

[] A Terminal Disclaimer and a check for [] \$55.00 (2814) [] \$110.00 (1814) to cover the requisite Government fee are also enclosed.

[] Also enclosed is _____.

[] Small entity status is hereby claimed.

[] Applicant(s) request continued examination under 37 C.F.R. § 1.114 and enclose the [] \$370.00 (2801) [] \$740.00 (1801) fee due under 37 C.F.R. § 1.17(e).

[] Applicant(s) previously submitted ___, on ___, for which continued examination is requested.

[] Applicant(s) request suspension of action by the Office until at least ___, which does not exceed three months from the filing of this RCE, in accordance with 37 C.F.R. § 1.103(c). The required fee under 37 C.F.R. § 1.17(i) is enclosed.

[] A Request for Entry and Consideration of Submission under 37 C.F.R. § 1.129(a) (146/246) is also enclosed.

[X] No additional claim fee is required.

[] An additional claim fee is required, and is calculated as shown below:

AMENDED CLAIMS					
	NO. OF CLAIMS	HIGHEST NO. OF CLAIMS PREVIOUSLY PAID FOR	EXTRA CLAIMS	RATE	ADDT'L FEE
Total Claims	31	MINUS 31 =	0	× \$18.00 (1202) =	0
Independent Claims	6	MINUS 6 =	0	× \$84.00 (1201) =	0
If Amendment adds multiple dependent claims, add \$280.00 (1203)					
Total Amendment Fee					
If small entity status is claimed, subtract 50% of Total Amendment Fee					
TOTAL ADDITIONAL FEE DUE FOR THIS AMENDMENT					0

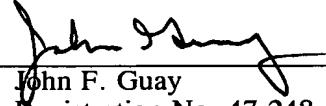
[] A claim fee in the amount of \$_____ is enclosed.

[] Charge \$_____ to Deposit Account No. 02-4800.

The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§ 1.16, 1.17, 1.20(d) and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800. This paper is submitted in duplicate.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: 
John F. Guay
Registration No. 47,248

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: November 13, 2002